Deep Learning - Week 10

1. Consider an input image of size $1000 \times 1000 \times 7$ where 7 refers to the number of channels (Such images do exist!). Suppose we want to apply a convolution operation on the entire image by sliding a kernel of size $1 \times 1 \times d$. What should be the depth d of the kernel?

   Correct Answer: 7

   Solution: To apply a convolution operation on an image of size $1000 \times 1000 \times 7$ with a kernel of size $1 \times 1 \times d$, the depth d of the kernel must match the number of channels in the input image.

   The number of channels in the input image is 7. For the convolution to be valid, the kernel's depth d should also be 7, as each kernel needs to process all channels of the input image simultaneously.

   Conclusion: The depth d of the kernel should be 7.

2. For the same input image in Q1, suppose that we apply the following kernels of differing sizes.

   $K_1 : 5 \times 5$
   $K_2 : 7 \times 7$
   $K_3 : 25 \times 25$
   $K_4 : 41 \times 41$
   $K_5 : 51 \times 51$

   Assume that stride $s = 1$ and no zero padding. Among all these kernels which one shrinks the output dimensions the most?

   (a) K_1
   (b) K_2
   (c) K_3
   (d) K_4
   (e) K_5

   Correct Answer: (e)
   Solution: To determine which kernel shrinks the output dimensions the most, we can calculate the output dimensions after applying each kernel. The formula for the output size of a convolution operation without padding and with a stride of 1 is:

$$\text{Output Size} = (\text{Input Size} - \text{Kernel Size}) + 1$$

   Given the input image size is 1000×1000 and stride $s = 1$, we can calculate the output dimensions for each kernel size.

   Kernel $K_1 : 5 \times 5$

$$\text{Output Size} = (1000 - 5) + 1 = 996$$

So, the output size will be 996×996.

Kernel $K_2 : 7 \times 7$

$$\text{Output Size} = (1000 - 7) + 1 = 994$$

So, the output size will be 994×994.

Kernel $K_3 : 25 \times 25$

$$\text{Output Size} = (1000 - 25) + 1 = 976$$

So, the output size will be 976×976.

Kernel $K_4 : 41 \times 41$

$$\text{Output Size} = (1000 - 41) + 1 = 960$$

So, the output size will be 960×960.

Kernel $K_4 : 51 \times 51$

$$\text{Output Size} = (1000 - 51) + 1 = 950$$

So, the output size will be 950×950.

Summary:

- $K_1 : 996 \times 996$
- $K_2 : 994 \times 994$
- $K_3 : 976 \times 976$
- $K_4 : 960 \times 960$
- $K_5 : 950 \times 950$

Among all these kernels, Kernel K_5 (51×51) shrinks the output dimensions the most, resulting in an output size of 950×950.

3. Which of the following is a technique used to fool CNNs in Deep Learning?

   (a) Transfer learning
   (b) Dropout
   (c) Batch normalization
   (d) Adversarial examples

Correct Answer: (d)
Solution: Adversarial examples are images that have been specifically designed to trick a CNN into misclassifying them. They are created by making small, imperceptible changes to an image that cause the CNN to output the wrong classification.

Transfer learning is a technique where a pre-trained model is fine-tuned on a new dataset to improve performance. It is not used to fool CNNs.

Dropout is a regularization technique used to prevent overfitting in neural networks.

Batch normalization is a method used to stabilize training by normalizing activations across mini-batches.

4. What is the motivation behind using multiple filters in one Convolution layer?

   (a) Reduced complexity of the network

   (b) Reduced size of the convolved image

   (c) Insufficient information

   (d) Each filter captures some feature of the image separately

   Correct Answer: (d)
   Solution: Increasing the number of filters at each layer creates more trainable parameters and increases the image's dimensions. However, we believe that each filter learns to capture some important image aspects. This is analogous to feature engineering in classical machine learning.

5. Which of the following statements about CNN is (are) true?

   (a) CNN is a feed-forward network

   (b) Weight sharing helps CNN layers to reduce the number of parameters

   (c) CNN is suitable only for natural images

   (d) The shape of the input to the CNN network should be square

   Correct Answer: (a),(b)
   Solution: Let's analyze each statement about Convolutional Neural Networks (CNNs):

   1. CNN is suitable only for natural images False. CNNs are not limited to natural images. They can be applied to a wide variety of data types such as medical images, time-series data (like EEG signals), audio signals (spectrograms), text (in word embeddings or character-level), and more. CNNs are effective whenever local spatial patterns or hierarchical features are important, regardless of the data type.

   2. The shape of the input to the CNN network should be square False. The input to a CNN does not have to be square. While many datasets like image data often use square inputs (e.g., 28×28 or 224×224), CNNs can handle rectangular inputs as well (e.g., 1280×720) as long as they maintain consistent height and width.

   3. CNN is a feed-forward network True. CNN is a type of feed-forward neural network. The information flows in one direction: from the input layer, through the convolutional and fully connected layers, to the output layer. There is no feedback or looping, as is common in recurrent neural networks (RNNs).

   4. Weight sharing helps CNN layers to reduce the number of parameters True. Weight sharing is a key feature of CNNs, particularly in convolutional layers. A single filter (kernel) is applied across the entire input image, and this filter's weights are shared across different locations. This significantly reduces the number of parameters compared to fully connected layers, where each weight is unique.

6. Consider an input image of size $100 \times 100 \times 1$. Suppose that we used kernel of size 3×3, zero padding $P = 1$ and stride value $S = 3$. What will be the output dimension?

(a) $100 \times 100 \times 1$

(b) $3 \times 3 \times 1$

(c) $34 \times 34 \times 1$

(d) $97 \times 97 \times 1$

Correct Answer: (c)

Solution: To calculate the output dimensions after applying a convolution operation, the formula is:

$$\text{Output size} = \left(\frac{\text{Input size} - \text{Kernel size} + 2P}{S}\right) + 1$$

Where:

- Input size is the spatial dimensions of the input image.
- Kernel size is the size of the kernel/filter.
- P is the amount of zero padding.
- S is the stride.

Given:

- Input size $= 100 \times 100 \times 1$ (we only care about the spatial dimensions, i.e., 100×100).
- Kernel size $= 3 \times 3$.
- Zero padding $P = 1$.
- Stride $S = 3$.

Let's calculate the output dimensions for both the height and width:

$$\text{Output size} = \left(\frac{100 - 3 + 2(1)}{3}\right) + 1$$

Simplifying:

$$\text{Output size} = \left(\frac{100 - 3 + 2}{3}\right) + 1 = \left(\frac{99}{3}\right) + 1 = 33 + 1 = 34$$

Therefore, the output dimensions are:

$$34 \times 34 \times 1$$

So, the output dimension is $34 \times 34 \times 1$.

7. Consider an input image of size $100 \times 100 \times 3$. Suppose that we use 8 kernels (filters) each of size 1×1, zero padding $P = 1$ and stride value $S = 2$. How many parameters are there? (assume no bias terms)

(a) 3

(b) 24

(c) 10

(d) 8

(e) 100

Correct Answer: (b)
Solution: To calculate the number of parameters in the convolutional layer, we need to consider the size of each kernel (filter) and the number of kernels. Let's break it down step by step:

Given Information:

- Input image size: $100 \times 100 \times 3$
- Number of kernels: 8
- Kernel size: 1×1
- Zero padding $P = 1$
- Stride $S = 2$

1. Number of Parameters per Kernel: Each kernel has a size of 1×1 and operates on all the input channels (3 channels for the input image). Therefore, the number of parameters in each kernel is:

$$\text{Parameters per kernel} = 1 \times 1 \times 3 = 3$$

2. Total Number of Parameters: Since there are 8 kernels, each with 3 parameters, the total number of parameters is:

$$\text{Total parameters} = 8 \times 3 = 24$$

8. What is the purpose of guided backpropagation in CNNs?

(a) To train the CNN to improve its accuracy on a given task.

(b) To reduce the size of the input images in order to speed up computation.

(c) To visualize which pixels in an image are most important for a particular class prediction.

(d) None of the above.

Correct Answer: (c)
Solution: Guided backpropagation is a technique used to visualize the parts of an input image that are most important for a particular class prediction. It achieves this by backpropagating the gradients of the output class with respect to the input image, but only allowing positive gradients to flow through the network.

9. Which of the following statements is true regarding the occlusion experiment in a CNN?

(a) It is a technique used to prevent overfitting in deep learning models.

(b) It is used to increase the number of filters in a convolutional layer.

(c) It is used to determine the importance of each feature map in the output of the network.

(d) It involves masking a portion of the input image with a patch of zeroes.

Correct Answer: (c),(d)
Solution: In the occlusion experiment, a patch of zeroes is placed over a portion of the input image to observe the effect on the output of the network. This helps to determine the importance of each region of the image in the network's prediction.

10. Which of the following architectures has the highest no of layers?

(a) AlexNet

(b) GoogleNet

(c) ResNet

(d) VGG

Correct Answer: (c)
Solution: Among the listed architectures, the one with the highest number of layers is ResNet.

Here's a brief comparison:

1. AlexNet: - Has 8 layers (5 convolutional layers followed by 3 fully connected layers).

2. GoogleNet (Inception v1): - Has 22 layers (not counting pooling layers).

3. ResNet: - Comes in different versions, with the most common being ResNet-50, ResNet-101, and ResNet-152. The number of layers for these are 50, 101, and 152 layers respectively, making it the deepest architecture on this list.

4. VGG: - VGG-16 has 16 layers, and VGG-19 has 19 layers.

Conclusion: ResNet, especially ResNet-152, has the highest number of layers among the architectures listed. Therefore, the answer is:

ResNet